**Homework 1205: A MIPS Program**

Last modified: 30 November 2021

Due: Tuesday, 7 December, 3 pm

**General Instructions**

For this assignment create a MIPS assembly source code program in a file with extension .s, containing only plain ASCII characters, with lines limited to 78 characters in length. Make sure you put your name in a header comment! You must complete this entirely in a Unix environment, either Linux or Mac OS.

Assembly style rules: lowercase hex letters, labels in column 1, instructions and directives in column 3, arguments in column 9, instruction comments in column 29, a comment on (almost) every instruction, a space after every comma, no camel case.

Pseudoinstructions should be used. Floating point instructions are not allowed.

Below is a C program that prompts the user for a string and a character and then determines whether the character is contained in the string, case insensitive. Your job is to convert this into an assembly language program. In general you should implement the assembly language exactly as the C is written, with a couple of exceptions. You may implement the variables string and character either as variables in main (on the stack) or as global variables (in the .data section). Also, the QtSpim read\_string syscall does not store the newline, so do not try to implement the line of C code with the strcspn call.

In your assembly code, pay particular attention to setting up and tearing down the stack frame of the functions, and to preserving and restoring those registers that need to be preserved. Use comments to clearly document which register is used to represent each parameter and variable.

The C program calls the library function toupper, but in our MIPS environment we do not have that function available. Therefore, you should assume you have this MIPS assembly version of toupper available, which you should include in your program exactly as written.

toupper:

# a0: the character parameter

# v0: the return value

addu $v0, $a0, $zero # copy a0 to v0

blt $v0, 0x61, tu\_end # 0x61 = 'a'; ignore chars less than

bgt $v0, 0x7a, tu\_end # 0x7a = 'z'; ignore chars greater than

subu $v0, $v0, 0x20 # 0x20 = 'a' - 'A'

tu\_end:

jr $ra

For your information, here is the C version of the toupper function:

char my\_toupper(char c)

{

if (c >= 'a')

{

if (c <= 'z')

{

c -= CASE\_DIFF; /\* convert a to uppercase \*/

}

}

return c;

}

Here is the C program for you to implement:

#include <ctype.h>

#include <stdio.h>

#include <string.h>

#define MAX\_CHARS 10

/\*\*

\* determine whether a string contains a character, each

\* case insensitive

\* @param str the string

\* @param ch the character

\* @return 1 if ch is in str; 0 otherwise

\*/

unsigned strcontainsi(const char \*s, char c);

int main(void)

{

char string[MAX\_CHARS + 1];

char character;

printf("Please enter a string: ");

fgets(string, MAX\_CHARS, stdin);

string[strcspn(string, "\n")] = '\0';

printf("Please enter a character: ");

character = (char)getchar();

if (strcontainsi(string, character))

{

printf("%s", string);

printf(" contains ");

printf("%c", character);

printf("\n");

}

else

{

printf("%s", string);

printf(" does not contain ");

printf("%c", character);

printf("\n");

}

return 0;

}

unsigned strcontainsi(const char \*str, char ch)

{

unsigned done = 0;

unsigned found = 0;

size\_t i = 0;

while (!done)

{

if (str[i] == '\0')

{

done = 1;

}

else if (toupper(str[i]) == toupper(ch))

{

found = 1;

done = 1;

}

else

{

i++;

}

}

return found;

}

By the due time, submit your source code to the [homework submission](https://borax.truman.edu/250/submit.php) page.